Transporte de Dicionário de Compressão

O Transporte de Dicionário de Compressão é uma forma de usar um dicionário de compressão compartilhado para reduzir drasticamente o tamanho do transporte de respostas HTTP.

Visão geral

Algoritmos de compressão são usados no HTTP para reduzir o tamanho dos recursos baixados pela rede, reduzindo o custo de largura de banda e o tempo necessário para carregar páginas. Algoritmos de compressão HTTP sem perdas funcionam encontrando redundância na fonte: por exemplo, lugares onde textos como a string "function" se repetem. Eles então incluem apenas uma cópia da string redundante e substituem ocorrências dela no recurso por referências a essa cópia. Como as referências são menores que a string, a versão comprimida é mais curta.

Por exemplo, considere este JavaScript:

Experimental: Esta é uma tecnologia experimental.
Verifique cuidadosamente a tabela de compatibilidade de navegadores antes de usar isso em produção.

Nota: Uma tentativa anterior dessa tecnologia foi chamada SDCH (Shared Dictionary Compression for HTTP), mas nunca teve amplo suporte e foi removida em 2017. O Transporte de Dicionário de Compressão é uma implementação melhor especificada e mais robusta, com consenso mais amplo da indústria.

Esse conteúdo poderia ser comprimido substituindo strings repetidas por referências a uma localização anterior e número de caracteres, como neste exemplo:

Neste exemplo, [0:9] refere-se à cópia dos 9 caracteres começando no caractere 0.
Note que este é um exemplo simplificado para demonstrar o conceito e que os algoritmos reais são mais complexos que isso.

Os clientes então podem reverter a compressão após o download para recriar o recurso original e não comprimido.

Dicionários de compressão

Algoritmos como a compressão Brotli e a compressão Zstandard alcançam ainda maior eficiência permitindo o uso de dicionários de strings comumente encontradas, de modo que você não precisa de cópias delas no recurso comprimido. Esses algoritmos vêm com um dicionário padrão predefinido que é usado ao comprimir respostas HTTP.

O Transporte de Dicionário de Compressão expande isso permitindo que você forneça seu próprio dicionário, o que é especialmente aplicável a um determinado conjunto de recursos. O algoritmo de compressão pode então referenciá-lo como uma fonte de bytes ao comprimir e descomprimir o recurso.

Assumindo que as referências do exemplo anterior estejam incluídas nesse dicionário comum, isso poderia ser ainda mais reduzido para:

```javascript
function b() {
  console.log("I am here");
}
```

```css
[d0:9]a[d10:20]Hello World![d42:46]
[d0:9]b[d10:20]I am here[d42:46]
```

O dicionário pode ser um recurso separado que é necessário apenas para o Transporte de Dicionário de Compressão, ou pode ser um recurso que o site já precisa de qualquer maneira.

Por exemplo, suponha que seu site use uma biblioteca JavaScript. Você normalmente carregaria uma versão específica da biblioteca, e poderia incluir o nome da versão no nome do arquivo da biblioteca, como:

```html
<script src="my-library.v1.js">
```

Quando o navegador carrega sua página, ele buscará uma cópia da biblioteca como um sub-recurso.

Se você então atualizar para a v2 da biblioteca, a maior parte do código da biblioteca provavelmente permanecerá igual. Assim, os sites podem reduzir muito o tamanho do download de my-library.v2.js dizendo ao navegador para usar my-library.v1.js como um dicionário de compressão para my-library.v2.js. Então, todas as strings que são comuns entre a v1 e a v2 não precisam ser incluídas no download da v2, porque o navegador já as possui. A maior parte do tamanho de download de my-library.v2.js será então apenas o delta entre as duas versões.

O Transporte de Dicionário de Compressão pode atingir uma ordem de magnitude maior em compressão do que a compressão utilizando um dicionário embutido padrão: veja os exemplos de transporte de dicionário de compressão para alguns resultados da vida real.

Formato de dicionário

Um dicionário de compressão não segue nenhum formato específico, nem possui um tipo MIME específico. Eles são arquivos regulares que podem ser usados na compressão de outros arquivos com conteúdo semelhante.

Versões anteriores de arquivos normalmente têm muito conteúdo similar, por isso são ótimos dicionários. Usar uma versão anterior de um arquivo como dicionário permite que o algoritmo de compressão referencie eficientemente todo o conteúdo que não foi alterado e capture apenas as pequenas diferenças na nova versão. Essa abordagem é chamada de **compressão delta**.

Outra abordagem é listar strings comuns (por exemplo, seus templates HTML) juntas em um novo arquivo dictionary.txt para que ele possa ser usado para comprimir páginas HTML no site. Você pode otimizar isso ainda mais usando ferramentas especializadas, por exemplo, o **gerador de dicionário do Brotli**, que reduz os dicionários até seu tamanho mínimo com sobreposição mínima.

Dicionários também podem ser usados para comprimir eficazmente formatos binários. Por exemplo, arquivos binários WASM são recursos grandes que também podem se beneficiar de compressão delta.

Recurso existente como um dicionário

Para usar um recurso como um dicionário, o servidor deve incluir o cabeçalho Use-As-Dictionary na resposta que fornece o recurso:

```http
Use-As-Dictionary: match="/js/app.*.js"
```

O valor deste cabeçalho indica quais recursos podem usar este recurso como dicionário: neste caso, isso inclui quaisquer recursos cujas URLs correspondam ao padrão fornecido.

Quando um recurso for solicitado mais tarde e corresponder ao padrão fornecido (por exemplo, app.v2.js), a requisição incluirá:

- um **hash SHA-256** do dicionário disponível no cabeçalho Available-Dictionary,
- juntamente com os valores dcb e/ou dcz no cabeçalho Accept-Encoding (para compressão delta usando Brotli ou ZStandard, conforme apropriado):

```http
Accept-Encoding: gzip, br, zstd, dcb, dcz
Available-Dictionary: :pZGm1Av0IEBKARczz7exkNYsZb8LzaMrV7J32a2fFG4=:
```

O servidor pode então responder com uma resposta codificada apropriadamente com a codificação de conteúdo escolhida indicada no cabeçalho Content-Encoding:

```http
Content-Encoding: dcb
```

Se a resposta for armazenável em cache (cacheable), ela **deve incluir um cabeçalho Vary** para evitar que caches sirvam recursos comprimidos com dicionário para clientes que não os suportam, ou que sirvam a resposta comprimida com o dicionário errado:

```http
Vary: accept-encoding, available-dictionary
```

Um id opcional também pode ser fornecido no cabeçalho Use-As-Dictionary, para permitir que o servidor localize mais facilmente o arquivo de dicionário, caso ele não seja armazenado pelo hash:

```http
Use-As-Dictionary: match="/js/app.*.js", id="dictionary-12345"
```

Se isso for fornecido, o valor será enviado em futuras requisições no cabeçalho Dictionary-ID:

```http
Dictionary-ID: "dictionary-12345"
```

O servidor ainda deve verificar o hash do cabeçalho Available-Dictionary — o Dictionary-ID é uma informação adicional para o servidor identificar o dicionário, mas **não substitui a necessidade** do cabeçalho Available-Dictionary.

Dicionário separado

Um documento HTML também pode fornecer um dicionário de compressão ao navegador que **não** é um recurso que o navegador já estaria baixando, usando um elemento como uma tag <script>. Existem dois métodos para fazer isso:

1. Incluir um elemento <link> cujo atributo rel esteja definido como compression-dictionary:

```html
<link rel="compression-dictionary" href="/dictionary.dat" />
```

2. Referenciar o dicionário usando o cabeçalho Link:

```http
Link: </dictionary.dat>; rel="compression-dictionary"
```

Esse dicionário é então baixado pelo navegador durante o tempo ocioso, e essa resposta **deve incluir** o cabeçalho Use-As-Dictionary:

```http
Use-As-Dictionary: match="/js/app.*.js"
```

A partir daí, o processo é semelhante ao exemplo anterior, quando um recurso correspondente é solicitado.

Criando respostas comprimidas com dicionário

Respostas comprimidas com dicionário podem usar os algoritmos Brotli ou ZStandard, com dois requisitos adicionais:

1. Elas **devem incluir um cabeçalho mágico** (magic header)
2. E o **hash embutido do dicionário**

Recursos comprimidos com dicionário podem ser criados dinamicamente, mas para recursos estáticos pode ser melhor criá-los com antecedência no momento da build. Ao usar versões anteriores como dicionários, será necessário decidir **quantas versões delta-comprimidas criar** — apenas para a versão anterior ou para as últimas X versões, para algum valor de X.

Exemplo com Brotli

Dado um arquivo de dicionário chamado dictionary.txt e um arquivo a ser comprimido chamado data.txt, o seguinte comando Bash comprimirá o arquivo usando Brotli, produzindo um arquivo comprimido chamado data.txt.dcb:

```bash
echo -en '\xffDCB' > data.txt.dcb && \
openssl dgst -sha256 -binary dictionary.txt >> data.txt.dcb && \
brotli --stdout -D dictionary.txt data.txt >> data.txt.dcb
```

Exemplo com ZStandard

Dado os mesmos arquivos de entrada, o seguinte comando Bash comprimirá o arquivo usando ZStandard, produzindo um arquivo chamado data.txt.dcz:

```bash
echo -en '\x5e\x2a\x4d\x18\x20\x00\x00\x00' > data.txt.dcz && \
openssl dgst -sha256 -binary dictionary.txt >> data.txt.dcz && \
zstd -D dictionary.txt -f -o tmp.zstd data.txt && \
cat tmp.zstd >> data.txt.dcz
```

Nota: Você precisará do OpenSSL instalado localmente, assim como Brotli ou ZStandard.

Restrições

Algoritmos de compressão estão sujeitos a riscos de ataques de segurança, portanto existem várias restrições para o **Transporte de Dicionário de Compressão**, incluindo:

- **Dicionários devem ser da mesma origem** que o recurso que os utiliza.
- Recursos comprimidos com dicionário devem ser da mesma origem da origem do documento, ou seguir as regras **CORS**, e portanto devem ser requisitados com o atributo crossorigin e servidos com um cabeçalho apropriado Access-Control-Allow-Origin.
- Dicionários estão sujeitos ao particionamento usual de cache HTTP e, portanto, **não podem ser compartilhados entre origens** mesmo que elas baixem os mesmos recursos. O dicionário precisará ser baixado novamente para cada origem.
- Além disso, os próprios dicionários **podem se tornar vetores de rastreamento**, então navegadores podem restringir esse recurso quando cookies estiverem desabilitados ou quando outras proteções adicionais de privacidade estiverem ativadas.

Especificações

Especificação:

- Transporte de Dicionário de Compressão (Compression Dictionary Transport)

Compatibilidade entre navegadores

As próximas páginas do documento apresentam tabelas de compatibilidade para os seguintes elementos e cabeçalhos:

- html.elements.link.rel.compression-dictionary
- http.headers.Accept-Encoding.dcb
- http.headers.Accept-Encoding.dcz
- http.headers.Available-Dictionary
- http.headers.Content-Encoding.dcb
- http.headers.Content-Encoding.dcz
- http.headers.Dictionary-ID
- http.headers.Use-As-Dictionary

As tabelas indicam:

- Suporte total
- Sem suporte
- Experimental (comportamento pode mudar no futuro)